AMENDMENTS TO THE CLAIMS

1. (Currently amended) A method for ~~solving a constraint satisfaction problem~~ testing a system, comprising:

receiving a set of variables that are characteristic of inputs to the system under test, the variables having respective input domains, and a set of relations among the variables;

building a network of one or more hyper-arcs representative of the set of relations, each hyper-arc corresponding to one of the relations and linking nodes in the network corresponding to the variables that are subject to the relation;

for each of the hyper-arcs, assembling the variables in a hierarchy based on the relation corresponding to the hyper-arc; and

reducing the input domains of the variables in the hierarchy, so as to determine respective output domains of the variables that are consistent with the relations; and

determining values of the inputs to be made to the system based on the output domains of the variables.

2. (Original) A method according to claim 1, wherein assembling the variables comprises arranging the variables in a hierarchical graph, having vertices corresponding to the variables.

3. (Original) A method according to claim 2, wherein arranging the variables in the hierarchical graph comprises arranging the graph so as to have the form of one or more trees.

4. (Original) A method according to claim 3, wherein reducing the input domains comprises reducing the input domains over each of the trees so as to find respective interim domains of the variables that are consistent with the relation over each of the trees, and combining the interim domains over all of the trees to determine the output domains.

5. (Original) A method according to claim 2, wherein receiving the set of relations comprises receiving a definition of the relations as a combination of operators, selected from a group of arithmetic, bitwise and logical operators, which are applied to the variables, and wherein arranging the variables in the graph

comprises inserting vertices in the graph corresponding to the operators, connecting the vertices corresponding to the variables.

6. (Original) A method according to claim 5, wherein reducing the input domains comprises finding projections of the operators onto the domains of the variables in the graph.

7. (Original) A method according to claim 6, wherein receiving the set of variables comprises receiving an output variable and at least one input variable for each of the operators, and

wherein finding the projections comprises projecting the domain of the at least one input variable of each of the operators onto the domain of the output variable thereof, and projecting the domain of the output variable of each of the operators onto the domain of the at least one input variable thereof.

8. (Original) A method according to claim 1, wherein building the network of the hyper-arcs comprises representing the set of relations as a disjunction of multiple relations, with one of the hyper-arcs corresponding respectively to each of the relations, and

wherein determining the respective output domains comprises determining interim domains of the variables for each of the hyper-arcs, and taking a union of the interim domains for each of the variables to determine the output domains.

9. (Original) A method according to claim 1, wherein reducing the input domains comprises determining the output domains such that for any given value in the respective output domain of each of the variables, there exist values of the other variables in the respective output domains thereof that, together with the given value, constitute a solution to the set of relations.

10. (Original) A method according to claim 1, wherein reducing the input domains comprises determining the output domains such that every set of values of the variables in the input domains that constitutes a solution to the set of relations is contained in the output domains of the variables.

11. (Original) A method according to claim 1, wherein receiving the set of relations comprises receiving a relation relating to at least three of the variables.

12. (Canceled)

13. (Currently amended) A method according to ~~claim 12~~ claim 1, wherein the system comprises an electronic processor, and wherein determining the values of the inputs comprises determining commands and addresses to be input to the processor.

14-18. (Canceled)

19. (Currently amended) A method for ~~solving a constraint satisfaction problem~~ testing a system, comprising:

receiving a set of variables that are characteristic of inputs to the system under test, the variables having respective input domains, and a set of constraints comprising a relation among at least three of the variables;

building a network of one or more hyper-arcs representing the constraints, the hyper-arcs comprising nodes representing the variables, one of the hyper-arcs corresponding to the relation among the at least three variables; and

reducing the input domains of the variables in the network of hyper-arcs, so as to determine respective output domains of the variables that are consistent with the set of constraints; and

determining values of the inputs to be made to the system based on the output domains of the variables.

20. (Original) A method according to claim 19, wherein reducing the input domains comprises finding projections of the relation onto the input domains of the variables.

21. (Original) A method according to claim 20, wherein receiving the set of constraints comprises receiving a definition of the relation as a combination of operators, selected from a group of arithmetic, bitwise and logical operators, which are applied to the variables, and wherein finding the projections comprises finding projections of the operators onto the domains of the variables.

22. (Original) A method according to claim 19, wherein reducing the input domains comprises determining the output domains such that for any given value in the respective output domain of each of the variables, there exist values of the other variables in the respective output domains thereof that, together with the given value, constitute a solution to the set of constraints.

23. (Original) A method according to claim 19, wherein reducing the input domains comprises determining the output domains such that every set of values of the variables in the input domains that constitutes a solution to the set of constraints is contained in the output domains of the variables.

24. (Currently amended) A method for ~~solving a constraint satisfaction problem~~ testing a system, comprising:

receiving a set of variables that are characteristic of inputs to the system under test, the variables having respective input domains, and a set of constraints comprising one or more relations defined as a combination of operators, selected from a group of arithmetic, bitwise and logical operators, which are applied to the variables;

building a network of one or more hyper-arcs representing the set of constraints, each hyper-arc corresponding to one of the relations expressed in terms of the operators and linking nodes in the network corresponding to the variables to which the operators are applied; and

reducing the input domains of the variables in the network responsive to the operators, so as to determine respective output domains of the variables that are consistent with the set of constraints; and

determining values of the inputs to be made to the system based on the output domains of the variables.

25. (Original) A method according to claim 24, wherein receiving the set of constraints comprises providing a language for specifying the constraints, the language having grammatical rules, and specifying the constraints using the language.

26. (Original) A method according to claim 24, wherein reducing the input domains comprises finding projections of the operators onto the domains of the variables.

27. (Original) A method according to claim 25, wherein receiving the set of variables comprises receiving an output variable and at least one input variable for each of the operators, and

wherein finding the projections comprises projecting the domain of the at least one input variable of each of the operators onto the domain of the output variable thereof, and projecting the domain of the output variable of each of the operators onto the domain of the at least one input variable thereof.

28. (Original) A method according to claim 24, wherein the operators comprise multi-variable operators, which receive two of more of the variables as their inputs.

29. (Original) A method according to claim 28, wherein the multi-variable operators comprise one or more operators selected from a group consisting of addition, subtraction, multiplication, division and modulo operators.

30. (Original) A method according to claim 28, wherein the multi-variable operators comprise one or more operators selected from a group consisting of an operator testing equality of two of the variables, an operator testing inequality of two of the variables, and an operator testing whether one of the variables is greater than another of the variables.

31. (Original) A method according to claim 28, wherein the multi-variable operators comprise one or more operators selected from a group consisting of a bitwise "and," bitwise "or" and bitwise "exclusive or" operations.

32. (Original) Apparatus for solving a constraint satisfaction problem, comprising a constraint processor, arranged to receiving a set of variables having respective input domains and a set of constraints comprising one or more relations among the variables, to build a network of one or more hyper-arcs representative of the set of constraints, each hyper-arc corresponding to one of the relations and linking nodes in the network corresponding to the variables that are subject to the relation and for each of the hyper-arcs, to assemble the variables in a hierarchy based on the relation corresponding to the hyper-arc, and to reduce the input domains of the variables in the hierarchy, so as to determine respective output domains of the variables that are consistent with the set of constraints.

33. (Original) Apparatus according to claim 32, wherein the hierarchy of the variables comprises a hierarchical graph, having vertices corresponding to the variables.

34. (Original) Apparatus according to claim 33, wherein the hierarchical graph has the form of one or more trees.

35. (Original) Apparatus according to claim 34, wherein the processor is arranged to reduce the input domains over each of the trees so as to find respective interim domains of the variables that are consistent with the relation over each of the trees,

and to combine the interim domains over all of the trees to determine the output domains.

36. (Original) Apparatus according to claim 33, wherein the set of constraints is defined as a combination of operators, selected from a group of arithmetic, bitwise and logical operators, which are applied to the variables, and wherein the graph comprises vertices corresponding to the operators, connecting the vertices corresponding to the variables.

37. (Original) Apparatus according to claim 36, wherein the processor is arranged to find projections of the operators onto the domains of the variables in the graph.

38. (Original) Apparatus according to claim 37, wherein the set of variables comprises an output variable and at least one input variable for each of the operators, and wherein the processor is arranged to project the domain of the at least one input variable of each of the operators onto the domain of the output variable thereof, and to project the domain of the output variable of each of the operators onto the domain of the at least one input variable thereof.

39. (Original) Apparatus according to claim 32, wherein the set of constraints is represented as a disjunction of multiple relations, with one of the hyper-arcs corresponding respectively to each of the relations, and wherein the processor is arranged to determine interim domains of the variables for each of the hyper-arcs, and to take a union of the interim domains for each of the variables to determine the output domains.

40. (Original) Apparatus according to claim 32, wherein the processor is arranged to determine the output domains such that for any given value in the respective output domain of each of the variables, there exist values of the other variables in the respective output domains thereof that, together with the given value, constitute a solution to the set of constraints.

41. (Original) Apparatus according to claim 32, wherein the processor is arranged to determine the output domains such that every set of values of the variables in the input domains that constitutes a solution to the set of constraints is contained in the output domains of the variables.

42. (Original) Apparatus according to claim 32, wherein the set of constraints comprises a relation that relates to at least three of the variables.

43. (Original) Apparatus according to claim 32, wherein the set of variables comprises variables that are characteristic of inputs to a system under test, and wherein the processor is arranged to determine values of the inputs to be made to the system based on the output domains of the variables.

44. (Original) Apparatus according to claim 43, wherein the system comprises an electronic device, and wherein the inputs comprise commands and addresses to be input to the device.

45. (Original) Apparatus according to claim 32, wherein the set of variables comprises control parameters of a mechanical system, and wherein the processor is arranged to generate a command to control the system based on the output domains of the parameters.

46. (Original) Apparatus according to claim 32, wherein the set of variables comprises features of an image containing visual information, and wherein the processor is arranged to identify an object in the image based on the features, responsive to the output domains.

47. (Original) Apparatus according to claim 32, wherein the set of variables comprises a natural language input, and wherein the processor is arranged to parse the natural language, responsive to the output domains, so as to interpret the language.

48. (Original) Apparatus according to claim 32, wherein the set of variables comprises characteristics of a condition, and wherein the processor is arranged to determine a diagnosis of the condition, based on the output domains.

49. (Original) Apparatus according to claim 32, wherein the set of variables comprises characteristics of resources whose use is to be scheduled, and wherein the processor is arranged to schedule the use of the resources subject to the set of constraints, based on the output domains.

50. (Original) Apparatus for solving a constraint satisfaction problem, comprising a constraint processor, arranged to receive a set of variables having respective input domains and a set of constraints comprising a relation among at least three of the variables, to build a network of one or more hyper-arcs representative of the set of

constraints, including a hyper-arc corresponding to the relation among the at least three variables and linking nodes in the network corresponding to the variables that are subject to the relation, and to reduce the input domains of the variables in the network of hyper-arcs, so as to determine respective output domains of the variables that are consistent with the set of constraints.

51. (Original) Apparatus according to claim 50, wherein the processor is arranged to determine the output domains by finding projections of the relation onto the input domains of the variables.

52. (Original) Apparatus according to claim 51, wherein the relation is defined as a combination of operators, selected from a group of arithmetic, bitwise and logical operators, which are applied to the variables, and wherein the projections comprise projections of the operators onto the domains of the variables.

53. (Original) Apparatus according to claim 50, wherein the processor is arranged to determine the output domains such that for any given value in the respective output domain of each of the variables, there exist values of the other variables in the respective output domains thereof that, together with the given value, constitute a solution to the set of constraints.

54. (Original) Apparatus according to claim 50, wherein the processor is arranged to determine the output domains such that every set of values of the variables in the input domains that constitutes a solution to the set of constraints is contained in the output domains of the variables.

55. (Original) Apparatus for solving a constraint satisfaction problem, comprising a constraint processor, arranged to receive a set of variables having respective input domains and a set of constraints comprising one or more relations defined as a combination of operators, selected from a group of arithmetic, bitwise and logical operators, which are applied to the variables, to build a network of one or more hyper-arcs representative of the set of constraints, each hyper-arc corresponding to one of the relations expressed in terms of the operators and linking nodes in the network corresponding to the variables to which the operators are applied, and to reduce the input domains of the variables in the network responsive to the operators, so as to

determine respective output domains of the variables that are consistent with the set of constraints.

56. (Original) Apparatus according to claim 55, wherein the constraints received by the processor are specified using a constraint-specification language having grammatical rules, and wherein the processor is arrange to build the network automatically based on the constraints specified in the language.

57. (Original) Apparatus according to claim 55, wherein the processor is arranged to reduce the input domains by finding projections of the operators onto the domains of the variables.

58. (Original) Apparatus according to claim 57, wherein the set of variables comprises an output variable and at least one input variable for each of the operators, and wherein the processor is arranged to project the domain of the at least one input variable of each of the operators onto the domain of the output variable thereof, and to project the domain of the output variable of each of the operators onto the domain of the at least one input variable thereof.

59. (Original) Apparatus according to claim 55, wherein the operators comprise multi-variable operators, which receive two of more of the variables as their inputs.

60. (Original) Apparatus according to claim 59, wherein the multi-variable operators comprise one or more operators selected from the group consisting of addition, subtraction, multiplication, division and modulo operators.

61. (Original) Apparatus according to claim 59, wherein the multi-variable operators comprise one or more operators selected from the group consisting of an operator testing equality of two of the variables, an operator testing inequality of two of the variables, and an operator testing whether one of the variables is greater than another of the variables.

62. (Original) Apparatus according to claim 59, wherein the multi-variable operators comprise one or more operators selected from a group consisting of a bitwise "and," bitwise "or" and bitwise "exclusive or" operations.

63. (Original) A computer software product for solving a constraint satisfaction problem, the product comprising a computer-readable medium in which program instructions are stored, which instructions, when read by a computer, cause the

computer, upon receiving a set of variables having respective input domains and a set of constraints comprising one or more relations among the variables, to build a network of one or more hyper-arcs representative of the constraint, each hyper-arc corresponding to one of the relations and linking nodes in the network corresponding to the variables that are subject to the relation and, for each of the hyper-arcs, to assemble the variables in a hierarchy based on the relation corresponding to the hyper-arc, and to reduce the input domains of the variables in the hierarchy, so as to determine respective output domains of the variables that are consistent with the set of constraints.

64. (Original) A computer software product for solving a constraint satisfaction problem, the product comprising a computer-readable medium in which program instructions are stored, which instructions, when read by a computer, cause the computer, upon receiving a set of variables having respective input domains and a set of constraints comprising a relation among at least three of the variables, to build a network of one or more hyper-arcs representative of the set of constraints, including a hyper-arc corresponding to the relation among the at least three variables and linking nodes in the network corresponding to the variables that are subject to the relation, and to reduce the input domains of the variables in the network of hyper-arcs, so as to determine respective output domains of the variables that are consistent with the set of constraints.

65. (Original) A computer software product for solving a constraint satisfaction problem, the product comprising a computer-readable medium in which program instructions are stored, which instructions, when read by a computer, cause the computer, upon receiving a set of variables having respective input domains and a set of constraints comprising one or more relations defined as a combination of operators, selected from a group of arithmetic, bitwise and logical operators, which are applied to the variables, to build a network of one or more hyper-arcs representative of the set of constraints, each hyper-arc corresponding to one of the relations expressed in terms of the operators and linking nodes in the network corresponding to the variables to which the operators are applied, and to reduce the input domains of the variables in the network responsive to the operators, so as to determine respective output domains of the variables that are consistent with the set of constraints.